- Introduction
- The Distance Machine
- Scales
- The ANN Library
- Work To Do
- Preliminary Performance Tests

Introduction

The Terabyte Analysis Machine is a cluster designed to allow research into the use of large scientific databases.

Our initial program is to explore efficient database re-indexing and re-partitioning as a way to take maximal advantage of sophisticated algorithms for use in specific astronomical problems.

The Distance Machine

As an example of a database re-indexing and re-partitioning in the TAM framework, the Distance Machine is an implementation of an analysis engine for the kth nearest neighbor search.

In astronomy, the kth nearest neighbor may be used to search for:

- Cluster of Galaxies
- Gravitational Lenses and Quasars
- Dwarf Galaxies

The Distance Machine uses the SDSS (SX) database

Scales

The SDSS database will have:

- 500 Gbytes database
- 250 million objects
- 100 million galaxies (example of the search for cluster of galaxies)
- 2.5 Kbytes object size
- 500 byte tag object size (example of tag with about 50 parameters of interest)
- 50 Gbytes tag database

Moving the tag database:

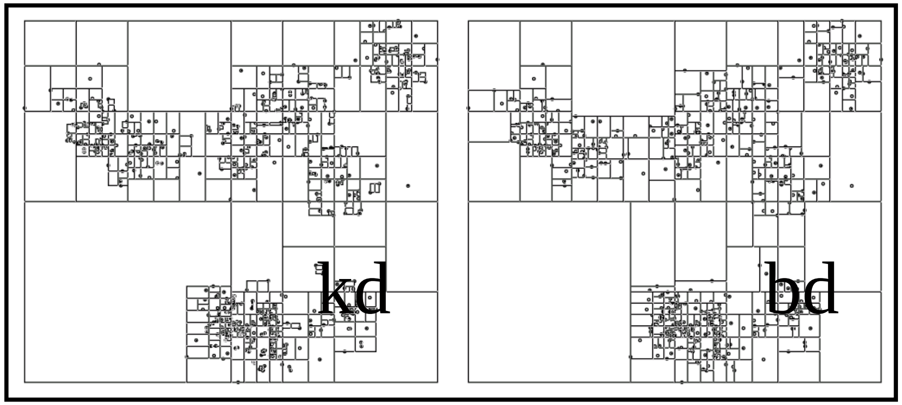
- 50 Gbytes at 10 Mbytes/sec => 1.5 hours to transfer over network
- 50 Gbytes at 30 Mbytes/sec => 0.5 hours to write to disk

Dimensionality of the problem:

- 120 parameters (total) per object
- 5 parameters used for cluster finding: ra, dec, g-r, r-i, and i'

The ANN Library

By **David M. Mount**, Department of Computer Science and Institute for Advanced Computer Studies, **University of Maryland**
www.cs.umd.edu/~mount/ANN/



ANN (Approximate Nearest Neighbor) is a C++ library that builds a tree structure from the data and search through it for the kth NN of any given query point.

ANN v0.2 works in memory only.
It has been modified to work with Objectivity.

Building a tree is a re-indexing of the data; this naturally yields to the re-clustering of the database.

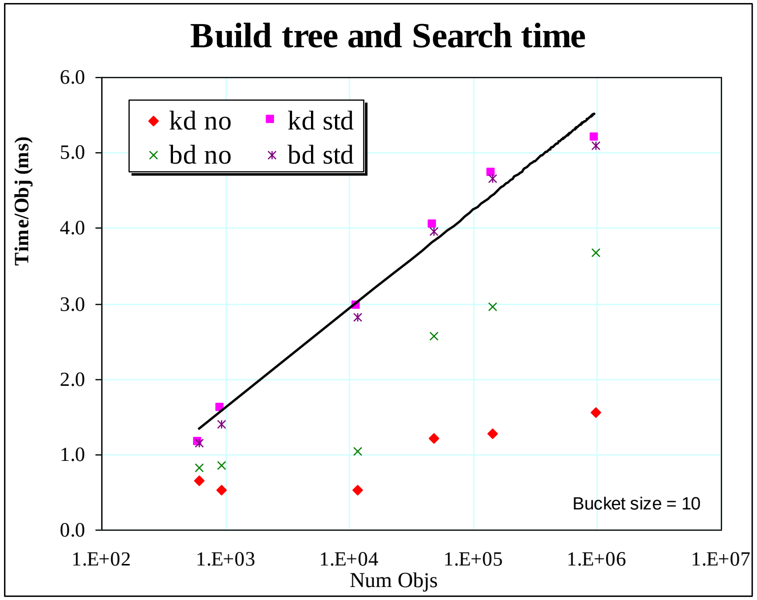
Work To Do

Caltech has collaborated to the project writing the module that extracts the tag objects from SX. This module has to be integrated in the TAM framework

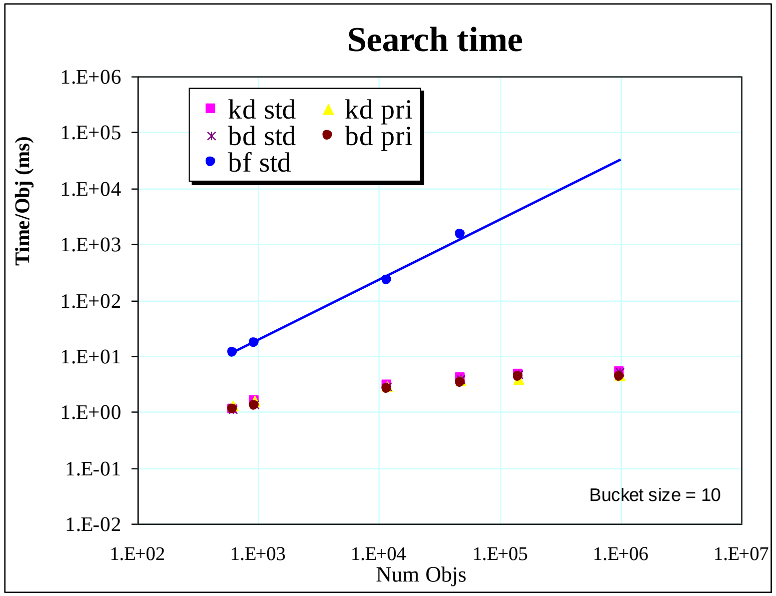
The module that re-clusters the database is designed and has to be implemented.

...then of course... a colorful cool demo...

Preliminary Performance Tests



I expect the tree based search time to be *O(N LogN)*. This hypothesis is true with a squared correlation coefficient (R^2) of 0.98



Comparison between the 3 different kinds of search algorithms. Standard (std) and Priority (pri) are tree based and *O(N LogN)*; Brute Force (bf) is the classical *O(N^2)* search algorithm.